Diamantopoulos, F. & Economides, A.A.: **A performance study of DSDV-based CLUSTERPOW and DSDV routing algorithms for sensor network applications**. *Proceedings IEEE International Symposium on Wireless Pervasive Computing 2006*.

A performance study of DSDV-based CLUSTERPOW and DSDV routing algorithms for sensor network applications.

Fotis Diamantopoulos and Anastasios A. Economides

Abstract— **Wireless adhoc sensor networks are ad hoc networks that consist of a number of autonomous, battery powered, static devices, communicating with each other through radio connections, using special routing algorithms. Many sensor network implementations use DSDV as their routing protocol. The wireless sensor networks' resources such as throughput and energy are scarce and need to be carefully used. Power control can be implemented by CLUSTERPOW algorithm. Among other factors that could waste the networks' resources and deplete the nodes' energy, is the routing protocol's overhead. DSDV is designed for mobile ad hoc networks and a large ratio of its traffic is generated to keep the routes updated. We studied the behavior of the protocols through simulation and found out that by carefully adjusting some parameters the performance improves, the routing overhead reduces and less energy is consumed.**

Index Terms— **CLUSTERPOW, DSDV, power control, power consumption, routing overhead, sensor networks**

I. INTRODUCTION

The advance of technology has enabled the creation of infrastructureless wireless networks, or wider known as ad hoc networks. Ad hoc networks can be categorized, based on their mobility, in mobile ad hoc networks (MANETs) and static or sensor ad hoc networks. Sensors are devices that can sense their environment, collect and often process data, depending on their size and cost. Usually the devices are small, connected through low bandwidth links that yield small data rates typically of a few kbps. Sensors include monitoring devices such as thermometers, barometers, and safety monitors such smoke detectors, or glass break detectors, and access control devices [1][2]. Many routing protocols have been developed specifically for sensor networks; most of them are designed for transmission of data to a central Base Station, while only a few of them support communication schemes like peer-to-peer or multicast [3].

There are however sensor applications that are designed with mobile ad-hoc routing protocols. Destination Sequenced Distance Vector (DSDV) [4] is a candidate routing algorithm for many sensor applications like the "Follow me" application that guides visitors to the location of a building or an application to assist workers in finding conference rooms [5]. Both applications could be also used in outdoor sites such as archaeological sites, where no infrastructure exists. Another application is the Multimedia Guidebook [6], which is based on sensors communicating through an Ethernet to provide multimedia information via Bluetooth to the user's mobile device. If the Ethernet is substituted with a wireless 802.11b network then the application can be deployed to outdoor archaeological and tourist sites, specially when the sites are expanding for areas of many km^2.

DSDV is a table-driven protocol. Each node's table contains all the network existing destinations, a next hop for every destination, and a metric that indicates the cost of the route. Also each destination has a sequence number, indicating how old a route is. When a route update with a higher sequence number is received, it replaces the old route. In case of different routes with the same sequence number, the route with the better metric is used. Updates have to be transmitted periodically or immediately when any significant topology change is detected.

CLUSTERPOW [7] is a power control algorithm that can be used with any routing algorithm. It presupposes that a network interface can transmit in several discrete power levels. An instance of the routing algorithm agent is active for each power level, so each level has its own routing table, in the case of a proactive algorithm. Thus a message can be sent using the lowest power level at which the destination is reachable. Power control is used to increase the network's capacity, decrease the contention of the link layer and save energy.

II. PRELIMINARIES

Ad-hoc routing protocols are divided in three main categories:

1. On-Demand or Reactive protocols, which construct only necessary routes on demand. The major representative protocols are AODV [8] and DSR [9].
2. Table-driven or proactive protocols, where each node maintains routing information for every possible destination. DSDV is the main representative.

Anastasios A. Economides is an Associate Professor in the Information Systems Department of the University of Macedonia (phone:+302310891799; e-mail: economid@uom.gr).

Diamantopoulos Fotis is a postgraduate student at the University of Macedonia, Thessaloniki, 54006, Greece (e-mail: diamand@csd.uoc.gr).

3. Hybrid protocols, which combine on-demand and proactive routing, like Zone Routing [10].

In general, on demand protocols are more preferable for high mobility, while proactive protocols like DSDV are suited for low mobility and static networks [11].

DSDV was developed by C. Perkins in 1994. Its primary design goals were to maintain simplicity, to solve the looping problem and to cope dynamically with network changes. Every node transmits its routing table every update interval or when triggered by a change in the topology, e.g. a new neighbour or a broken link. When receiving an update a node will wait for a "settling" period before forwarding it, in case it receives a better or a newer route.

However in a static or sensor network topology changes are rarely happening. Topology could change only in cases like hardware failure, depletion of energy or radio interference from an external source. Thus a lot of overhead may be wasteful when the algorithm tries to keep routes updated in order to support mobility. In DSDV, this is accomplished by the routing update interval. Moreover the number of nodes in a sensor network is often very large. However, DSDV does not support scalability. Simulation studies [12][13], which have been carried out for different proactive protocols show high levels of data throughput and significantly less delays than on-demand protocols (such as DSR) only for networks made up of up to 50 nodes. Therefore, in small networks running real-time applications (e.g. video conferencing), where low end-to-end delay is highly desirable, proactive routing protocols may be more beneficial, but as the number of the nodes increases, either the algorithm has to be modified to improve its performance or another algorithm must be used.

CLUSTERPOW is a power control algorithm that belongs to a family of power control algorithms along with COMPOW, LOADPOW and MINPOW [7]. In CLUSTERPOW, each node runs a routing protocol daemon at each power level. In the case of a proactive protocol, it independently builds a routing table for every power level by exchanging hello messages at only that power level. To forward a packet for a destination, a node consults the lowest power routing table in which the destination is present, and forwards the packet at the minimum power level to the next hop.

As we will show, this algorithm suffers from the amount of produced overhead in dense networks with a large number of nodes. Authors consider overhead based on the average number of neighbours of a common wireless network [14] but they don't proceed to a full analysis. Suppose each routing daemon broadcasts one hello message, of which each routing entry are *b* bytes, every *T* seconds. Having *l* power levels, n_i neighbour nodes per level, and *N* total network nodes, each node would receive an overhead of

$$B = \sum_{i=1}^{l} \frac{b \cdot N \cdot n_i}{T} \text{ bytes/sec (1)}$$

In a network of 10 nodes with approximately 6 neighbour nodes within every level's range and 6 power levels and a 100 bytes routing entry resulting in a 1000 bytes message sent every 5 seconds, the total consumed bandwidth would be 7200 bytes/sec or 60 kbps, that is 3% of a 2Mbps bandwidth. This is the argument of CLUSTERPOW's authors. However in the case of a homogeneous network in an area of A m^2 with N nodes each transmitting with a range R, the average number of neighbours within range R$_i$ would be

$$n_i = \frac{N \cdot \pi \cdot R_i^2}{A} \text{ nodes} \qquad (2)$$

For an area of 0.25 km^2 with 100 nodes, each node transmitting with a range of 250 meters, we have an average of 78.5 nearby nodes. Substituting (2) in (1) yields

$$B = \sum_{i=1}^{l} \frac{N^2 \cdot \pi \cdot R_i^2 \cdot b}{AT} \text{ bytes/sec.} \quad (3)$$

This equation is true also for the case of one power level, i.e. DSDV. Additionally there are triggered updates, most of which are transmitted during the initialisation of the network, when the construction of the routing tables is taking place. A model that calculates both triggered and periodic overhead is given in [15]: Let *h* be the average frequency of triggered routing updates, *S* the size of the periodically broadcast table, Δ the average neighbours of each node, i.e. eq. (2) for a homogenous network. If *E* denotes the average number of emissions to achieve a topology broadcast, we denote by *o* the broadcast optimization factor, i.e., $o = E/N$, ($1/\Delta \le o \le 1$), then the consumed bandwidth of every transmission level *l* is:

$B_l = h_l \cdot b \cdot N + o_l \cdot S \cdot N^2/T$ bytes/sec. (5)

Knowing that $S=b*N$ the total bandwidth consumed is

$$B = \sum_{i=1}^{l} (\mathrm{h}_i \cdot \mathrm{b} \cdot \mathrm{N} + \frac{\mathrm{o}_i \cdot \mathrm{b} \cdot \mathrm{N}^3}{T}) \text{ bytes/sec. (6)}$$

However this equation is not taking into account of any dropped routing packets, retransmissions etc. Therefore to understand the actual impact of the routing update interval to the network performance and validate our model we have to resort to simulation.

III. RELATED WORK

The routing overhead of both proactive and reactive protocols is examined in many publications [11,12,13,15]. A mathematical model is given in [15] as well, describing overhead under mobility and immobility, though it does not describe the parameters of different transmission levels and the size of the routing tables that we consider. Modifications to the DSDV protocol have been proposed that manage to dynamically adjust the update interval according to the network mobility.

The ARM-DSDV [16] protocol has two controls. The update-period control maintains the mobility metric, based on the rate of change in its neighbourhood, i.e., the set of nodes within radio range, and dynamically adjusts the routing update period. The update-content control maintains the route-demand metric and dynamically adjusts the content of routing updates, sending regularly updates only for the most recently used routes and sparsely for the rest.

In DREAM [13], routing overhead can be reduced by making the rate at which route updates are sent analogous to the speed at which each node travels. Minimum Displacement Update Routing (MDUR) [17] attempts to disseminate route update packet information to the network when they are required rather than using purely periodic updates. This is achieved by setting the updating rate proportional to the distance a node moves. The rate of displacement can be measured using a Global Positioning System (GPS).

Another modification of DSDV is the Fisheye State Routing (FSR) that sends updates to its nearby nodes more frequently than to its distant nodes [18].

Randomized-DSDV [19] randomizes the routing interval according to a routing probability distribution so that it eliminates the broadcast storm of simultaneous updates.

There is little or no literature that clarifies the impact of the routing update interval to a static network performance.

IV. METHOD OF SIMULATION

We have run simulations with scenarios using DSDV based CLUSTERPOW, and DSDV. For our simulations we have used ns2, the Network Simulator [21] with Vikas Kawadia's modifications for CLUSTERPOW [20]. We modified the code so that we can simulate and measure the energy consumption of the algorithms. Initially we have simulated a random network of 120 static nodes in a 600m*600m area, 20 nodes of which communicate by sending CBR packets that cross the whole network topology. Each scenario was run with different random topologies.

As studied in [22] the average consumption of a Lucent IEEE 802.11 network interface is 1400mW in transmit mode, 1100 mW in receive mode and 830 mW in idle. We followed this model but with a few modifications. There has not been any research on the consumption of IEEE 802.11 network interfaces transmitting in a discrete number of power levels, on the contrary with [23] that examines power consumption of low-rate and low power sensors. We also simulated the rates and consumption of a sensor node that uses the magnitude of values given in [23]. Details of the second set of simulations are given in Table 2.

The transmission power for a node to achieve a transmission range of 250m is 281mW. So deducting it from consumed power, the circuitry and initialisation power consumption for transmission remains, which is 1.119 W. We add the signal power for each transmission level to the circuitry and initialisation power. That may not be very accurate but it serves well enough for our purpose, since we would like to understand the average magnitude of the consumption and not to have exact results, since not all network interfaces exhibit the same consumption as well.

We investigated the time needed to discover all the routes of the network and complete the tables, and found that it was independent of the routing update interval, due to the fact that changes in topology trigger routing updates. For the 0.36 km^2 area containing 120 nodes, the average time to complete the routing tables was 180 seconds for CLUSTERPOW - DSDV and 130 seconds for DSDV. When nodes began to communicate before the tables were complete, the performance was very poor with a very high packet loss ratio. That happens because the greatest amount of control traffic is generated during the discovery of the routes, which congests the network. Therefore we considered a warm up period of 180 seconds before nodes began sending their packets.

TABLE 1
SIMULATION PARAMETERS

SIMULATOR	NS2 v2.26
SIMULATION TIME	1000s
TRAFFIC	Constant Bit Rate UDP, TCP
MAC	IEEE 802.11
LINK DATA RATE	2 Mbps
NUMBER OF CLUSTERPOW POWER LEVELS	6
TRANSMISSION RANGE PER POWER LEVEL	250, 210, 170, 130, 90, 50 meters
TRANSMISSION POWER PER LEVEL	281 mW, 140mW, 60mW, 20 mW, 4.73mW, 0.45mW
TRANSMIT POWER DRAIN	1.119 W + Transmission Power
RECEIVE POWER DRAIN	1 W
IDLE POWER DRAIN	0.83 W
ROUTING PROTOCOL WARM-UP TIME	180 sec
TRIGGERED UPDATE SETTLING PERIOD	6 sec

TABLE 2
SIMULATION PARAMETERS

SIMULATOR	NS2 v2.26
SIMULATION TIME	100000s
MAC	IEEE 802.11
LINK DATA RATE	20 kbps
NUMBER OF CLUSTERPOW POWER LEVELS	6
TRANSMISSION RANGE PER POWER LEVEL	150, 110, 85, 68, 52, 38 meters
TRANSMISSION POWER PER LEVEL	36.3 mW, 10mW, 3.42mW, 1.32 mW, 0.437mW, 0.117mW
TRANSMIT POWER DRAIN	0.071 W + 10*Transmission Power
RECEIVE POWER DRAIN	0.051 W
IDLE POWER DRAIN	0.027 W
INITIAL ENERGY	2000 Joule
TRIGGERED UPDATE SETTLING PERIOD	6 sec

V. SIMULATION RESULTS

The simulation results are shown in the following figures (Figures 1-8). In order to discover the influence of the routing update interval to the total network time (that is the time until the first node of the network runs out of battery) we configured each node with a 500 joule initial energy.

The results show a significant improvement of the performance when we used a routing update interval of 60 seconds and more. The overhead of CLUSTERPOW with a 60 seconds interval is only the 18% of the overhead when we use a 15 seconds interval, while with intervals from 60 seconds

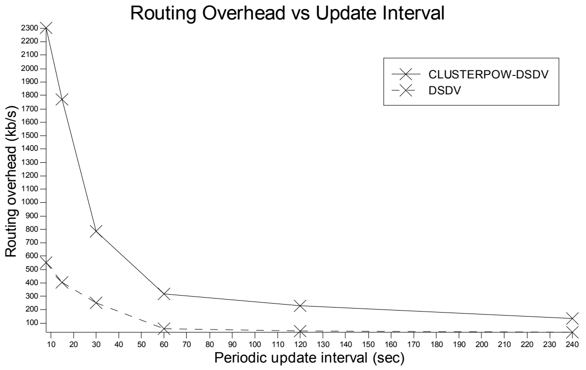

Fig.1. Routing Overhead (in kilobytes per second) in relation to the Routing Update Interval (in seconds)

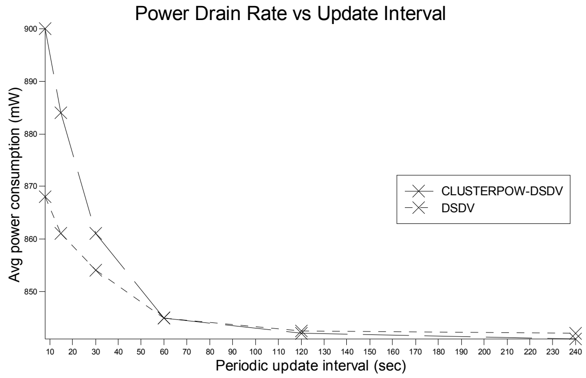

Fig. 2. Power Drain Rate (in milliwatts) in relation to the Routing Update Interval

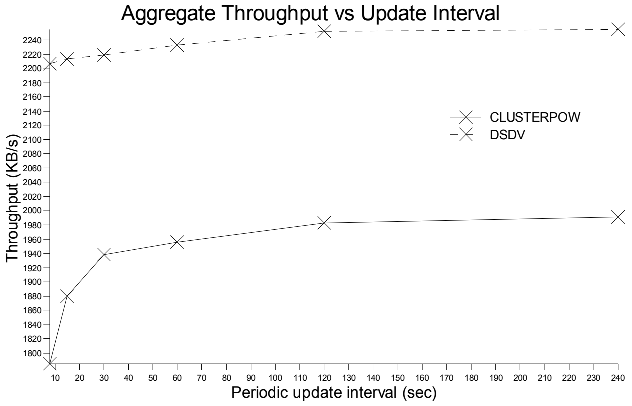

Fig. 3. Aggregate throughput in relation to the Routing Update Interval

and bigger, DSDV exhibits only a 10% of the 15 seconds interval overhead. That is consistent with eq. (6).

Power consumption is also analogous to the overhead, and so it decreases when the update interval increases, having an impact to the network time. Also we have to remark that the power consumption per byte is different in the power control algorithm, because not every byte is transmitted with the same power level. So while CLUSTERPOW overhead production is greater, its power consumption converges with the consumption of DSDV as the interval acquires a duration of 60sec, after which CLUSTERPOW becomes more energy efficient. This fact is reflected in fig. 5 depicting the total network time, where CLUSTERPOW is more effective at intervals longer than 120 seconds.

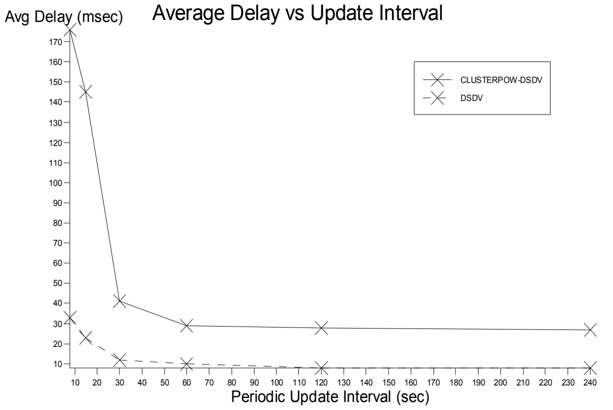

Fig.4 End-to-end Average Delay in relation to the Routing Update Interval

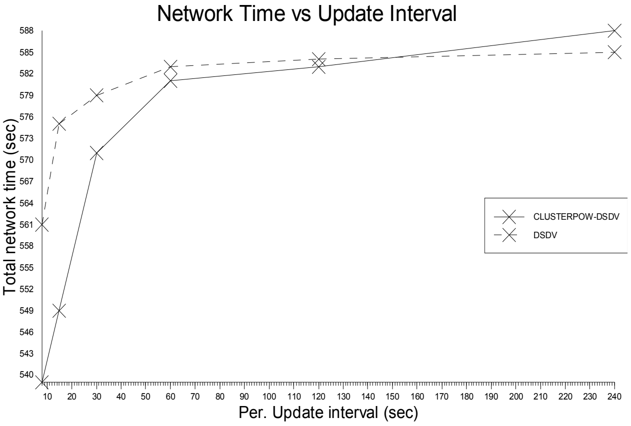

Fig.5. Total Network Time (time until the first node runs out of energy) opposed to the Periodic Update Interval

The average end-to-end delay decreases when the interval increases, since the smaller interval is producing bigger amounts of traffic that translate to more collisions, retransmissions and packet drops. Additionally DSDV improves 4% its throughput, however CLUSTERPOW has a 11% improvement, comparing the results of the 8sec interval to the 240sec.

The results show that performance is proportional to the inverse of the update interval, confirming eq. (6). We ran more simulations modifying some of the parameters in order to verify our hypothesis. We began simulating an area of 400×400 m^2 with 20 nodes, increasing the nodes by 20 in each simulation. No other communication was exchanged between the nodes. Figures 6 and 7 show the results, in contrast with the graphic depiction of eq. (6). Each table entry has a size of 16 bytes for DSDV and 32 bytes for CLUSTERPOW, and the total time of the simulation is 1000s.

These results verify eq (6) i.e. that overhead is inversely proportional to the update interval. The grey lines in fig. 5 depict the equation's solution setting the value of o=0.07 for 20 nodes, o=0.039 for 40 nodes, o=0.027 for 60 nodes and o=0.021 for 80 nodes, while h=1/12 (since 6 sec is the update "settling" time) in all cases. If o_t is the sum of the optimisation

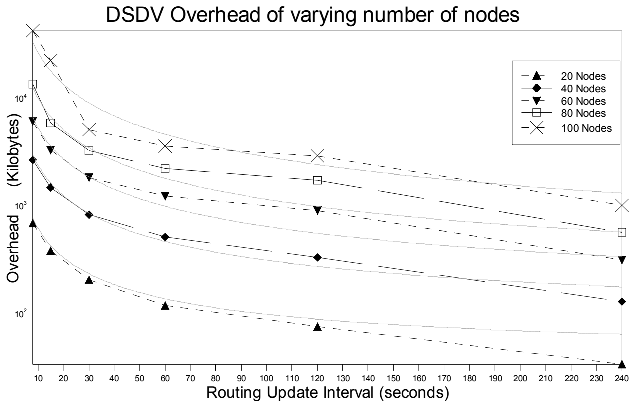

Fig 6. Simulation results of the overhead produced from five networks, each with a different number of nodes. The grey line next to each simulation result depicts equation (6) graphic representation for the corresponding number of nodes.

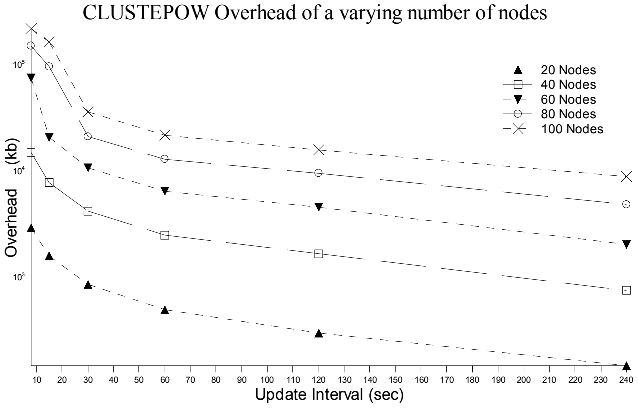

Fig 7. Simulation results for CLUSTERPOW for different numbers of nodes

factor of every transmission power level, then CLUSTERPOW overhead follows $B=o_t\cdot N\hat{}3\cdot b/T+h\cdot b\cdot N$. However *o* depends also on the average number of neighbours of the nodes. The values of *o* given above produce the most accurate results when they are of the form:

$o=\pi\cdot R^2\cdot\sqrt{\log(N)}/A\cdot N$. (7)

Equation (6), however does not produce accurate results when routing data are being transmitted simultaneously with the application data, due to the interference, collisions and retransmissions. Fig 9. shows the influence of the network load on overhead. Therefore overhead production is much higher than expected and we should examine all possible ways to minimise it.

While the ratio of the decrease of the overhead traffic in relation to the update frequency is high, the ratio of the increase of the total network time and the power save is not that impressive. We have a 5.5 times reduction in the overhead in CLUSTERPOW and a 10 times reduction in DSDV, when changing from a 15 seconds interval to a 60 seconds one, but the increase in the network time is only 7% for CLUSTERPOW and 0.8% for DSDV. This happens because the power consumption of the idle state is 0.83 Watts, and most of the energy is spent in idle time.

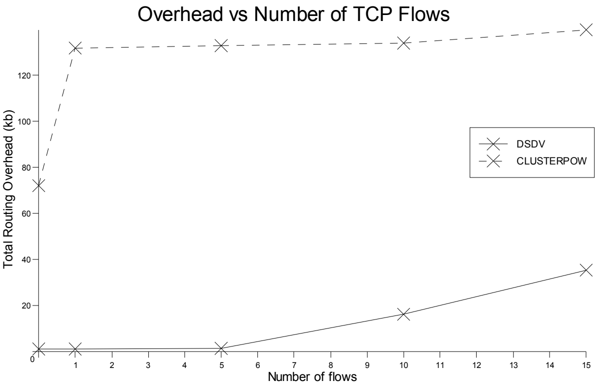

Figure 8. Overhead production at various network load levels.

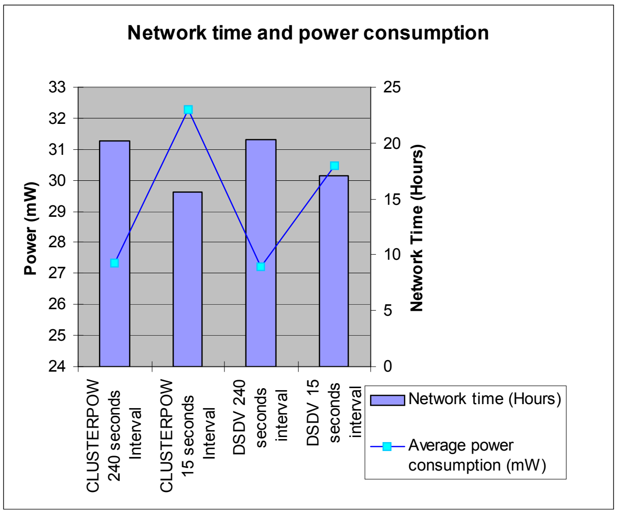

Fig. 9. The total network time of a sensor network with an initial energy of 2000J per node.

We also ran simulations of a network of nodes that approach the lower power consumption characteristics of the sensor nodes [23]. The details are given in Table 2. The energy is given by a typical 1 cm^3 battery, as stated in [24] that examines the volumetric characteristics of sensors power sources. We simulated 40 nodes in an 800×800 area. The results are given in fig. 10. The values of the network time and the power consumption in this case are much more impressive. Using a 240 seconds interval with CLUSTERPOW provides about 5 extra hours of total network time from a total of 15 hours of operation when using a 15 seconds interval. There is a 25% saving of the total the network time and a 16% average power saving. With DSDV we have a 3 hours and 20 minutes longer network time using a 240 second interval than the 17 hours network time of the 15 seconds interval and a 10% power consumption save.

The first conclusion we can draw from our results is that when using a proactive routing protocol in a static network, routing updates should be adjusted with the minimum possible update frequency, e.g. at a node's hardware failure, at node addition, at external interference. Of course issues like scalability are not solved by merely decreasing the update frequency, but with different proactive protocols (e.g. FSR) where the periodic update should be performed as rarely as possible.

We saw that with a typical node configuration the feasible time of the network was a few hours, but there are nodes with even better characteristics that can last for days. However most of the power saving is achieved, apart from minimizing the traffic as possible, by inducing nodes to sleeping states. In this case, the update frequency should be small, but not smaller than the frequency active nodes change.

Finally modifying the update interval could benefit applications for quasi-static ad-hoc networks, i.e. networks with very limited mobility. As such would be a conference session during which all the participants are sitting. Since the mobility is limited, the update interval could be lengthened as much as possible.

VI. CONCLUSION

We have exposed that routing algorithms designed mostly for mobile ad hoc networks, produce unnecessary traffic when they are used for static and sensor ad hoc networks, even for quasi-static networks. We studied the amount of the overhead created in DSDV and CLUSTERPOW algorithms and also have discovered its relationship with the routing update interval. We have run our simulation in a dense network where the results would be clearer and we have discovered that the produced overhead is analogous to the update interval frequency. A very small update frequency manages to reduce overhead, network latency and power consumption to a very satisfactory level. This conclusion can be used in combination with other power saving techniques to minimise the power consumption. These results show us how significant is to study all the details of algorithms in the circulating bibliography, and how we can make the best use of it.

VII. FUTURE WORK

Many studies could be performed to see how the routing interval or other protocols' mechanisms could be used for higher performance in specific applications, e.g. for video transferring or QoS in static networks, also what level of scalability could we achieve. We should also simulate the combination of proactive algorithms with power saving techniques to find out the highest possible savings from the modified parameters of the protocol.

REFERENCES

[1] I. F. Akyildiz et al., "Wireless Sensor Networks: a survey", *Computer Networks*, Vol. 38, pp. 393-422, March 2002.

[2] H. Karl and A. Willig, "A short survey of wireless sensor networks", Technical Report TKN-03-018, Telecommunication Networks Group, Technical University Berlin, October 2003.

[3] J. N. Al-Karaki, and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", IEEE Wireless Communications, vol. 11, Issue 6, pp. 6-28, Dec. 2004

[4] C.E. Perkins, and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", Comp. Comm. Rev., pp. 234-244, Oct. 1994.

[5] W. S. Conner, J. Heidemann, L. Krishnamurthy, Xi Wang, and M. Yarvis, "Workplace Applications of Sensor Networks," In Wireless Sensor Networks: A Systems Perspective, Nirupama Bulusu and Sanjay Jha (eds.), Artech House, August 2005 (to appear).

[6] M. Ros, M. D'Souza, M. Chan, K. Bialkowski, A. Postula, N. Bergmann, and A. Toth, "Using Wireless Sensors as Selection Devices for a Multimedia Guidebook Scenario", In Proc. RealWSN 2005: Workshop on Real-World Wireless Sensor Networks, Stockholm, Sweden, 20-21 June 2005.

[7] V. Kawadia, and P. R. Kumar, "Power control and clustering in ad hoc networks", In Proceedings of the IEEE INFOCOM Conference, pages. 459–469, 2003.

[8] C.E. Perkins, "Ad hoc on demand distance vector (AODV) routing", IETF Internet Draft (November 1998) http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-02.txt

[9] D.B. Johnson, and D.A. Maltz, "Dynamic source routing in ad hoc wireless networks", in: Mobile Computing, eds. T. Imielinski and H.F. Korth, chapter 5, pp. 153–181. Kluwer Academic, 1996.

[10] Z. J. Hass, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", IETF, Internet Draft, draft-zone-routing- protocol-00.txt, November 1997.

[11] P. Karavetsios & A.A. Economides, "Performance comparison of distributed routing algorithms in ad hoc mobile networks", WSEAS Transactions on Communications, Vol. 3, Issue 1, pp. 317-321, 2004.

[12] J. Broch, D. A. Maltz, D. B. Johnson, Y-Ch Hu, and J. Jetcheva. "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", In ACM Mobile Computing and Networking, pages 85–97, Oct. 1998.

[13] S. Basagni, I. Chlamtac, V.R. Syrotivk, and B.A. Woodward. "A Distance Effect Algorithm for Mobility (DREAM)", In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'98)*, Dallas, TX, 1998.

[14] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," Proceedings of European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications. pp. 156-162, Feb. 25-28, 2002. Florence, Italy.

[15] Viennot L.,Jacquet P., Clausen T.H.,"Analyzing Control Traffic Overhead versus Mobility and DataTraffic Activity in Mobile Ad-Hoc Network Protocols", Wireless Networks Vol.10,pp. 447–455, 2004,Kluwer Academic Publishers.

[16] S. Ahn, and A. U. Shankar, "Adapting to route-demand and mobility in ad hoc network routing", Computer Networks, Volume 38, Issue 6 , April 2002, 745 – 764.

[17] M. Abolhasan1, and T. Wysocki1, "Displacement-based Route Update Strategies for Proactive Routing Protocols in Mobile Ad Hoc Networks". *Workshop on the Internet, Telecommunications and Signal Processing* (WITSP 2003), Coolangatta, Gold Coast, Australia, Dec 8-11, 2003.

[18] M. Gerla, X. Hong, and G. Pei, "Fisheye State Routing Protocol (FSR) for Ad Hoc Networks," http://www.ietf.org/proceedings/01mar/I-D/manet-fsr-00.txt, IETF MANET Working Group-Internet Draft, Dec. 2001.

[19] A. Boukerche, and S. K. Das: Congestion Control Performance of R-DSDV Protocol in Multihop Wireless Ad Hoc Networks", Wireless Networks, Volume 9, Number 3, May 2003, pp. 261-270 (2003), Springer Science + Business Media B.V.

[20] [V. Kawadia, "Transmit Power Control Protocols for Ad Hoc Networks" http://www.ir.bbn.com/~vkawadia/txpower.html

[21] The NS2 Network Simulator, University of Southern California / Information Sciences Institute –http://www.isi.edu/nsnam/ns

[22] Feeney, L. M., and Nilsson, M. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. Proceedings of IEEE Conference on Computer Communications (IEEE InfoCom), Anchorage AK, USA, April 2001.

[23] Vijay Raghunathan, Curt Schurgers, Sung Park, Mani B. Srivastava, "Energy-Aware Wireless Microsensor Networks," *IEEE Signal Processing Magazine*, Vol.19, No.2, pp. 40-50, March 2002.

[24] Shad Roundy, Dan Steingart, Luc Frechette, Paul K. Wright, Jan M. Rabaey: "Power Sources for Wireless Sensor Networks." Wireless Sensor Networks, First European Workshop EWSN 2004: pp. 1-17.